Claims

What is claimed is:

1. A method for performing an exclusion join of at least a first table T1 and a second table T2, where each of the tables has an associated Star Map, S1 and S2, respectively, and each Star Map includes bitmap entries having locations indexed by the hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates the presence of one or more rows in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the method including:

a) performing one or more Boolean operations using the bitmap entries of the Star Maps S1 and S2 to produce bitmap entries in a Star Map SJ where there is a corresponding set bitmap entry in S1 and no corresponding set bitmap entry in S2;

b) using SJ to identify qualifying rows from the first table T1; and

c) providing the qualifying rows as an exclusion join result.

2. The method of claim 1, wherein using SJ to identify qualifying rows from the first table T1 includes performing the following until all rows from T1 have been selected:

b1) selecting a row in T1;

b2) hashing the combined entries in the one or more join key columns of the selected row to identify a bitmap entry in SJ; and

b3) if the identified bitmap entry in SJ is set, identifying the selected row as a qualifying row.

3. The method of claim 2, wherein the exclusion join includes join conditions and using SJ to identify qualifying rows from the first table T1 further includes:

b4) if the identified bitmap entry in SJ is not set, attempting to find a row in T2 that has entries in its one or more join key columns that together hash to the location of the identified unset bitmap entry in SJ; and

b5) if no such row can be found or if no found row from T2 satisfies the join conditions, identifying the selected row from T1 as a qualifying row.

4. The method of claim 1, further comprising

only performing b and c if the number of unset bitmap entries in SJ is less than a threshold.

5. The method of claim 4, wherein the threshold is related to the number of rows in T2 and a number of blocks used to store T2.

6. The method of claim 1, further comprising

determining an expected join cardinality; and

only performing a, b and c if the expected join cardinality is less than a cardinality threshold.

7. The method of claim 1, wherein one of the one or more Boolean operations produces an intermediate Star Map SINT.

8. The method of claim 7, wherein the intermediate Star Map SINT results from ANDing Star Maps S1 and S2 and Star Map SJ results from exclusive ORing Star Map SINT and S1.

9. The method of claim 7, wherein the intermediate Star Map SINT results from ORing Star Maps S1 and S2 and Star Map SJ results from exclusive ORing Star Map SINT and S2.

10. The method of claim 7, wherein the intermediate Star Map SINT results from inverting Star Map S2 and Star Map SJ results from ANDing Star Maps SINT and S1.

11. The method of claim 1, wherein a plurality of tables T1' and tables T2' and a plurality of associated Star Maps S1' and Star Maps S2' are provided, whereby Star Maps S1' are logically ORed to generate Star Map S1 and Star Maps S2' are logically ORed to generate Star Map S2.

12. A computer program, stored on a tangible storage medium, for performing an exclusion join of at least a first table T1 and a second table T2, where each of the tables has an associated Star Map, S1 and S2, respectively, and each Star Map includes bitmap entries having locations indexed by the hash of one or more of values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates the presence of one or more rows in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the program including executable instructions that cause a computer to:

a) perform one or more Boolean operations using the bitmap entries of the Star Maps S1 and S2 to produce bitmap entries in a Star Map SJ where there is a corresponding set bitmap entry in S1 and no corresponding set bitmap entry in S2;

b) use SJ to identify qualifying rows from the first table T1; and

c) provide the qualifying rows as an exclusion join result.

13. The computer program of claim 12, wherein the executable instructions for causing a computer to use SJ to identify qualifying rows from the first table T1 cause the computer to perform the following until all of the rows of T1 have been selected:

b1) select a row in T1;

b2) hash the combined entries in the one or more join key columns of the selected row to identify a bitmap entry in SJ; and

b3) if the identified bitmap entry in SJ is set, identify the found row as a qualifying row.

14. The computer program of claim 13, wherein the exclusion join includes join conditions, and the executable instructions for causing a computer to use SJ to identify qualifying rows from the first table T1 cause the computer to:

b4) if the identified bitmap entry in SJ is not set, attempt to find a row in T2 that has entries in its one or more join key columns that together hash to the location of the identified unset bitmap entry in SJ; and

b5) if no such row can be found or if no found row from T2 satisfies the join conditions, identify the selected row from T1 as a qualifying row.

15. The computer program of claim 12, where the computer:

only performs b and c if the number of unset bitmap entries in SJ is less than a threshold.

16. The computer program of claim 15, where the threshold is related to the number of rows in T2 and a number of blocks used to store T2 rows.

17. The computer program of claim 12, where the computer further

determines an expected join cardinality; and

only performs a, b and c if the expected join cardinality is less than a cardinality threshold.

18. The computer program of claim 12, wherein an intermediate Star Map SINT is generated.

19. The computer program of claim 18, wherein the intermediate Star Map SINT results from ANDing Star Maps S1 and S2 and Star Map SJ results from exclusive ORing Star Map SINT and S1.

20. The computer program of claim 18, wherein the intermediate Star Map SINT results from ORing Star Maps S1 and S2 and Star Map SJ results from exclusive ORing Star Map SINT and S2.

21. The computer program of claim 18, wherein the intermediate Star Map SINT results from inverting Star Map S2 and Star Map SJ results from ANDing Star Maps SINT and S1.

22. The computer program of claim 12, wherein a plurality of tables T1' and tables T2' and a plurality of associated Star Maps S1' and Star Maps S2' are provided, whereby Star Maps S1' are logically ORed to generate Star Map S1 and Star Maps S2' are logically ORed to generate Star Map S2.

23. A database system for accessing a database, the database system including

a massively parallel processing system including

one or more nodes;

a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs;

a plurality of virtual processes each of the one or more CPUs providing access to one or more processes;

each process configured to manage data stored in one of a plurality of data-storage facilities;

at least a first table T1 and a second table T2, being distributed among the data-storage facilities;

each of the tables having an associated Star Map, S1 and S2, respectively, each Star Map being distributed among the data-storage facilities,

each Star Map including bitmap entries having locations indexed by the hash of one or more of values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates the presence of one or more rows in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry;

a join process executed on one or more of the plurality of CPUs that cause the CPUs to

a) perform one or more Boolean operations using the bitmap entries of the Star Maps S1 and S2 to produce bitmap entries in a Star Map SJ where there is a corresponding set bitmap entry in S1 and no corresponding set bitmap entry in S2;

b) use SJ to identify qualifying rows from the first table T1; and

c) provide the qualifying rows as an exclusion join result.

24. The database system of claim 23, wherein when using SJ to identify qualifying rows from the first table T1 the computer repeats the following until all rows from T1 have been selected:

b1) selects a row in T1;

b2) hashes the combined entries in the one or more join key columns of the selected row to identify a bitmap entry in SJ; and

b3) if the identified bitmap entry in SJ is set, identifies the selected row as a qualifying row.

25. The database system of claim 24, wherein the exclusion join includes join conditions, and when using SJ to identify qualifying rows from the first table T1 the computer further performs the following:

b4) if the identified bitmap entry in SJ is not set, attempts to find a row in T2 that has entries in its one or more join key columns that together hash to the location of the identified unset bitmap entry in SJ; and

b5) if no such row can be found or if no found row from T2 satisfies the join conditions, identifies the selected row from T1 as a qualifying row.

26. The database system of claim 23, where the computer further:

only performs b and c if the number of unset bitmap entries in SJ is less than a threshold.

27. The database system of claim 26, wherein the threshold is related to the number of rows in T2 and a number of blocks used to store T2 rows.

28. The database system of claim 23, where the computer further

determines an expected join cardinality; and

only performs a, b and c if the expected join cardinality is less than a cardinality threshold.

29. The database system of claim 23, in which, when performing the one or more Boolean operations, the CPUs generate an intermediate Star Map SINT.

30. The database system of claim 29, wherein the intermediate Star Map SINT results from ANDing Star Map S1 and S2 and Star Map SJ results from exclusive ORing Star Map SINT and S1.

31. The database system of claim 29, wherein the intermediate Star Map SINT results from ORing Star Map S1 and S2 and Star Map SJ results from exclusive ORing Star Map SINT and S2.

32. The database system of claim 29, wherein the intermediate Star Map SINT results from inverting Star Map S2 and Star Map SJ results from ANDing Star Map SINT and S1.

33. The database system of claim 23, wherein a plurality of tables T1' and tables T2' and a plurality of associated Star Maps S1' and Star Maps S2' are provided, whereby Star Maps S1' are logically ANDed to generate Star Map S1 and Star Maps S2' are logically ORed to generate Star Map S2.

34. The database system of claim 33 where each of tables T1' and T2' are indexed by the hash of the combined entries in its respective one or more join key columns, and when using SJ to identify qualifying rows from the tables T1' the computer uses the hash value associated with the location of an unset bitmap entry in SJ as an index to retrieve rows from tables T2'.

35. The database system of claim 23 where one or more of the Star Maps is a table and a first portion of the hash value that indexes the locations of a Star Map defines a row within the Star Map and a second portion of the hash value defines an offset within the defined row.

36. The database system of claim 23 where the first portion is the first half of the hash value and the second portion is the second half of the hash value.

37. The database system of claim 23 where the hash value is 32 bits long, the first portion is the first 16 bits of the hash value, and the second portion is the second 16 bits of the hash value.

38. The database system of claim 23 where each entry in a Star Map is one bit.

39. The database system of claim 23 where each entry in a Star Map is sixteen bits.

40. The database system of claim 23 where each Star Map entry includes one or more bits and each bit corresponds to the hash of one or more values associated with the one or more join key columns of its associated table.

41. A method for performing an exclusion join of at least a first table T1 and a second table T2, where table T2 has an associated Star Map S2 which includes bitmap entries having locations indexed by the hash of one or more values associated with one or more join key columns of T2, where a bitmap entry in S2, if set, indicates the presence of one or more rows in T2 that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the method including:

a) selecting a row in T1;

b) hashing the combined entries in the one or more join key columns of the selected row to identify a bitmap entry in S2;

c) if the identified bitmap entry in S2 is not set, identifying the selected row as a qualifying row;

d) if the identified bitmap entry in S2 is set, finding one or more rows in T2 that have entries in its one or more join key columns that together hash to the location of the identified set bitmap entry in S2;

e) if none of the one or more found rows from T2 satisfy the join conditions, identifying the selected row from T1 as a qualifying row;

f) repeating a)-e) until all of the rows in T1 have been selected; and

g) providing the qualifying rows as an exclusion join result.

42. A method for performing a join of at least a first table T1 and a second table T2, where each of the tables has an associated Star Map, S1 and S2, respectively, and each Star Map includes bitmap entries having locations indexed by the hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates the presence of one or more rows in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the method including:

a) if the bitmap entry corresponding to the location in S1 identified by the hash of a NULL is set, probe T1 for a row having NULLs in its one or more join key columns, and if such a row is found, exiting the query with a "no rows found" message;

b) otherwise if the bitmap entry corresponding to the location in S2 identified by the hash of a NULL is set, probe T2 for a row having NULLs in its one or more join key columns, and if such a row is found, exiting the query with a "no rows found" message; and

c) otherwise perform the join operation.